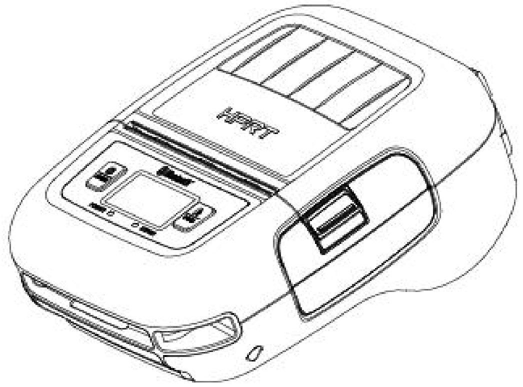
Mobile Printer

HM-Z3 & HM-T3
Programming Manual



REVISION RECORDS

REV.	DATE	DESCRIPTION	Drawn	Checked	Approved
1.0	2017.04.10	——	Wu Zhenrong	Yang kaiqu	Lin yang
1.1	2017.12.22	A.Modify the format of commands with “!”	Yan Xiaomei	Lin yang	Lin yang
1.2	2018.04.28	A.Add descriptions to command ‘TEXT’	Yan Xiaomei	Lin yang	Lin yang
1.3	2018.07.10	A.Add descriptions of ‘front’ to command ‘TEXT’ B.Add command ‘BKT’	Zhou Shulan	Lin yang	Lin yang
1.4	2018.11.29	A.Add descriptions to command ‘TEXT’	Chen Hua	Lin yang	Lin yang
1.5	2019.01.31	A.Add command POPRINT	Chen Hua	Lin yang	Lin yang

Table of Contents

1. INTRODUCTION

This manual details the various commands in the CPCL language which enable the programmer to utilize the built in text, graphics, bar code printing and communications capabilities of mobile printers.

The following notation conventions are used throughout this manual:

{ } Required item

[ ] Optional item

( ) Abbreviated command

< > Literal item

A space character is used to delimit each field in a command line.

Many commands are accompanied by examples of the command in use. After the word "Input" in each

example, the set of commands are displayed followed by a sample printout ("Output")resulting from the printer processing those commands.

2. PRINTER COMMANDS

A label file always begins with the "!" character followed by an "x" offset parameter, "x" and "y" axis resolutions, a label length and finally a quantity of labels to print. The line containing these parameters is referred to as the Command Start Line.

A label file always begins with the Command Start Line and ends with the "PRINT" command. The commands that build specific labels are placed between these two commands.

A space character is used to delimit each field in a command line.

Note: Every line in the command session must be terminated with both carriage-return and line-feed characters. All Printer Commands must be in uppercase characters ONLY.

Printer Commands

Format:

<!> {offset} <200> <200> {height} {qty}

where:

<!>: Use '!' to begin a control session.

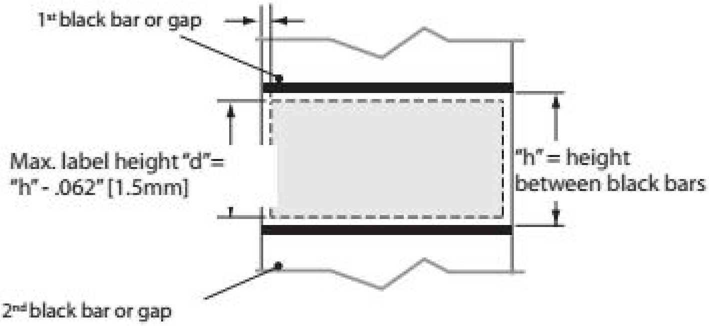
{offset}:The horizontal offset for the entire label. This value causes all fields to be offset horizontally by the specified number of UNITS.

<200>:Horizontal resolution (in dots-per-inch).

<200>:Vertical resolution (in dots-per-inch).

{height}:The maximum height of the label.

The maximum label height is calculated by measuring from the bottom of the first black bar (or label gap) to the top of the next black bar (or label gap). Then 1/16" [1.5mm] is subtracted from this distance to obtain the maximum height. (In dots: subtract 12 dots on 203 d.p.i printers; 18 dots on 306 d.p.i. printers)


1st black bar or gap
Max. label height "d"= "h" - .062" [1.5mm]
"h" = height between black bars
2nd black bar or gap

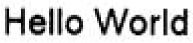
{qty}: Quantity of labels to be printed. Maximum = 1024.

Printer Command Example

Input

```
! 0 200 200 210 1
TEXT 4 0 30 40 Hello World
FORM
PRINT
```

Output



Hello World

PRINT Command

The PRINT command terminates and prints the file. This must always be the last command (except when in Line Print Mode). Upon execution of the PRINT command, the printer will exit from a control session. Be sure to terminate this and all commands with both carriage-return and line-feed characters.

Format:
{command}
where:
{command}: PRINT

END Command

The END command properly terminates a command and executes it without printing.

Format: END
{command}
where:
{command}: END
Example:

```
REM Send a blank label
! 0 200 200 240 1\r\n
PAGE-WIDTH 240\r\n
BOX 0 0 200 200 10\r\n
BOX 50 50 220 220 10\r\n
END\r\n
```

ABORT Command

The ABORT command terminates a current control session without printing.

Format:
{command}
where:
{command}: ABORT

FORM Command

The FORM command instructs the printer to feed to top of form after printing.

Format:
{command}
where:
{command}: FORM

In the following example, the printer will execute a form feed after the label is printed. See the SETFF (set form feed) command in the section on designing receipts and lists for information on setting printer behavior when the FORM command is executed.

Example
Input:

```
! 0 200 200 3 1
IN-CENTIMETERS
CENTER
TEXT 4 1 0 .5 Form Command
FORM
PRINT
```

UNITS Commands

The units commands are used to specify a measurement system for all subsequent command fields in a control session. Coordinates, widths, and heights for all control commands can be entered with precision to four decimal places. By placing a units command immediately after the first line in a control session, the specified measurement system will also apply to the offset and height fields. The printer measurement system will default to dots until a units command is issued.

Format:

{command}

where:

{command}: Choose from the following:

IN-INCHES Measurement in inches.

IN-CENTIMETERS Measurement in centimeters.

IN-MILLIMETERS Measurement in millimeters.

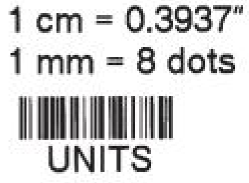
IN-DOTS Measurement in dots.

UNITS Examples

Input 1 :

```
! 0.3937 200 200 1 1
IN-INCHES
T 4 0 0 0 1 cm = 0.3937”
IN-DOTS
T 4 0 0 48 1 mm = 8 dots
B 128 1 1 48 16 112 UNITS
T 4 0 48 160 UNITS
FORM
PRINT
```

Output 1:

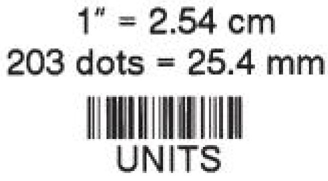


1 cm = 0.3937”
1 mm = 8 dots
UNITS

Input 2

```
! 0 200 200 2.54 1
IN-CENTIMETERS
T 4 0 1 0 1” = 2.54 cm
IN-MILLIMETERS
T 4 0 0 6 203 dots = 25.4 mm
B 128 0.125 1 6 12 14 UNITS
T 4 0 16 20 UNITS
FORM
PRINT
```

Output 2



1” = 2.54 cm
203 dots = 25.4 mm
UNITS

POPRINT Command

The poprint command is used to print for rotating 180°.

The function is same to PRINT command, the difference is that it prints all the content in rotation of 180°.

This command will not affect the next label.

USING COMMENTS

Comments can be added between the first line of a command session and the “PRINT” command.

A comment is placed in the file by starting a line with the ‘;’ character in the first column. Any remaining text to the end of the line will be ignored. Comments are illegal between the CONCAT and ENDCONCAT commands.

Comments Example

Input:

```
! 0 200 200 25 1
IN-MILLIMETERS
JOURNAL
; Center justify text
CENTER
; Print the words ‘A COMMENT’
TEXT 5 1 0 5 A COMMENT
; Print the label and go to top of next form
FORM
PRINT
```

Output:

A COMMENT

3.TEXT

TEXT Commands

The TEXT command is used to place text on a label. This command and its variants control the specific font number and size used, the location of the text on the label, and the orientation of this text. Standard resident fonts can be rotated in 90˚ increments as shown in the example.

Format:

{command} {font} {size} {x} {y} {data}

where:

{command}: Choose from the following:

{command}	Result
TEXT (or T)	Prints text horizontally.
VTEXT (or VT)	Prints text (vertically) rotated 90 degrees counterclockwise.
TEXT90 (or T90)	(Same as VTEXT above.)
TEXT180 (or T180)	Prints text (upside down) rotated 180 degrees counterclockwise.
TEXT270 (or T270)	Prints text (vertically) rotated 270 degrees counterclockwise.

{font}: Name/number of the font.

{size}: Size identifier for the font.(Default is 0, maximum is 7)

{x}: Horizontal starting position.

{y}: Vertical starting position.

(x, y) is the coordinate withing the current page.

{data}: The text to be printed.

Font Parameter Setting:

Font Size	Dots
1	24dots*24 dots
2	24 dots *24 dots
3	20 dots *20 dots
4	32 dots *32 dots
5	24 dots *24 dots
7	24 dots *24 dots
8	24 dots *24 dots
20	16 dots *16 dots
55	16 dots *16 dots
28	28 dots *28 dots

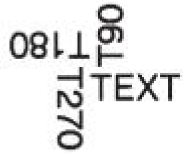
Example

Input:

```
! 0 200 200 210 1
TEXT 4 0 200 100 TEXT
```

```
TEXT90 4 0 200 100 T90
TEXT180 4 0 200 100 T180
TEXT270 4 0 200 100 T270
FORM
PRINT
```

Output:



TEXT Concatenation Commands (CONCAT and VCONCAT)

Text concatenation allows you to assign different character styles to strings, printing them with uniform spacing on the same text line. This command can be used in combination with scalable fonts. See Scalable Concatenation Commands

Format:

{command} {x} {y} {font} {size} {offset} {data} “ “ “ “ {font} {size} {offset} {data} <ENDCONCAT>

where:

{command}: Choose from the following:

CONCAT: Horizontal concatenation.

VCONCAT: Vertical concatenation.

{x}: Horizontal starting position.

{y}: Vertical starting position.

{font}: Name/number of the font.

{size}: Size identifier for the font.

{offset}: Unit-value to offset text from the starting position. Used to align individual text strings or create superscript/subscript characters.

{data}: Text to be printed.

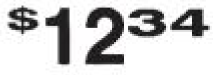
<ENDCONCAT>: Terminates concatenation.

Text Concatenation Example

Input:

```
! 0 200 200 210 1
CONCAT 75 75
4 2 5 $
4 3 0 12
4 2 5 34
ENDCONCAT
FORM
PRINT
```

Output:



MULTILINE (ML) Commands (For HM-T3 only)

MULTILINE (ML) allows you to print multiple lines of text using the same font and line-height.

Format:

```
{command} {height}
{text} {font} {size} {x} {y}
{data}
  “
{data}
<ENDMULTILINE>
```

where:

{command}: MULTILINE (or ML)- Prints multiple lines of text.

{height}: Unit-height for each line of text.

{text}: Text command (TEXT, VTEXT, etc.).

{font}: Name/number of the font.

{size}: Size identifier for the font.

{x}: Horizontal starting position.

{y}: Vertical starting position.

{data}: Text to be printed.

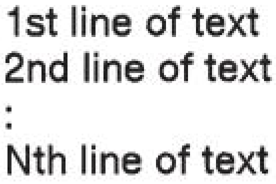
<ENDMULTILINE> (or ENDML): Terminates MULTILINE.

ML Command Example

Input:

```
! 0 200 200 210 1
ML 47
TEXT 4 0 10 20
1st line of text
2nd line of text
:
Nth line of text
ENDML
FORM
PRINT
```

Output:



```
1st line of text
2nd line of text
:
Nth line of text
```

COUNT Command

The COUNT command is used for printing multiple labels where a numeric text field or numeric data encoded in a bar code is to be incremented or decremented for each label. The TEXT/BARCODE command string must contain this numeric data as the last characters of the string. The numeric data portion can be up to 20 characters, and can be preceded by the '-' sign. Incrementing or decrementing the numeric data thru '0' is not allowed. Leading zeros will be retained. Up to three COUNT commands can be used in a label file.

The numeric data incremented/decremented is contained in the TEXT or BARCODE command that immediately preceded the COUNT command.

Format:

{command} {numeric value}

where:

{command}: COUNT

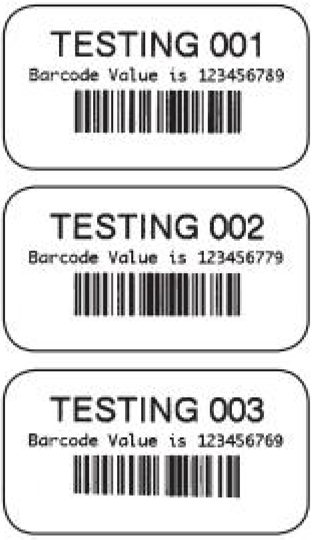
{numeric value}: Any integer value up to 20 characters. The value can be preceded by a '-' sign if decrementing of the TEXT/BARCODE value is desired. Leading zeros will be retained in the output.

COUNT Command Example

Input:

```
! 0 200 200 210 3
; Print 3 labels
CENTER
TEXT 4 0 0 50 TESTING 001
COUNT 1
TEXT 7 0 0 100 Barcode Value is 123456789
COUNT -10
BARCODE 128 1 1 50 0 130 123456789
COUNT -10
FORM
PRINT
```

Output:



TESTING 001
Barcode Value is 123456789

TESTING 002
Barcode Value is 123456779

TESTING 003
Barcode Value is 123456769

SETMAG Command

The SETMAG command magnifies a resident font to the magnification factor specified.

Format:

{command} {w} {h}

where:

{command}: SETMAG

{w}: Width magnification of the font. Valid magnifications are 1 thru 16.

{h}: Height magnification of the font. Valid magnifications are 1 thru 16.

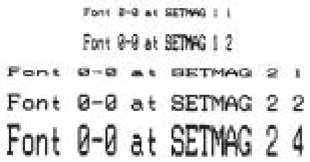
NOTE: The SETMAG command stays in effect after printing a label. This means that the next label printed will use the most recently set SETMAG values. To cancel any SETMAG values and allow the printer to use its default font sizes, use the SETMAG command with magnifcations of 0,0.

SETMAG Command Example

Input:

```
! 0 200 200 210 1
CENTER
SETMAG 1 1
TEXT 0 0 0 10 Font 0-0 at SETMAG 1 1
SETMAG 1 2
TEXT 0 0 0 40 Font 0-0 at SETMAG 1 2
SETMAG 2 1
TEXT 0 0 0 80 Font 0-0 at SETMAG 2 1
SETMAG 2 2
TEXT 0 0 0 110 Font 0-0 at SETMAG 2 2
SETMAG 2 4
TEXT 0 0 0 145 Font 0-0 at SETMAG 2 4
; Restore default font sizes
SETMAG 0 0
FORM
PRINT
```

Output:



```
Font 0-0 at SETMAG 1 1
Font 0-0 at SETMAG 1 2
Font 0-0 at SETMAG 2 1
Font 0-0 at SETMAG 2 2
Font 0-0 at SETMAG 2 4
```

BKT Command

BKT command is used to add the watermark on the label. This command and its generated commands can control the detailed font size , text position and text direction on the label. The standard usual font can rotate in 90° increment, shown as below.

Format:

{command}{font} {size} {x} {y} {data}

Among which:

{command}： select from the below table

{command}	Result
BKTEXT (or BKT)	Prints text horizontally.
VBKTEXT (or VBKT)	Prints text (vertically) rotated 90 degrees counterclockwise.
BKTEXT90 (or BKT90)	(Same as VTEXT.)
BKTEXT180 (or BKT180)	Prints text (upside down) rotated 180 degrees counterclockwise.
BKTEXT270 (or BKT270)	Prints text (vertically) rotated 270 degrees counterclockwise.

{font}： font name/number

{size}： font size

{x}： horizontal start position

{y}： vertical start position

{data}： text to print

{font} parameter setting

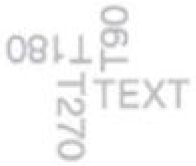
Font	Size
1	24dots*24 dots
2	24 dots *24 dots
3	20 dots *20 dots
4	32 dots *32 dots
5	24 dots *24 dots
7	24 dots *24 dots
8	24 dots *24 dots
20	16 dots *16 dots
55	16 dots *16 dots
28	28 dots *28 dots

Example

Input:

```
! 0 200 200 210 1
TEXT 4 0 200 100 TEXT
TEXT90 4 0 200 100 T90
TEXT180 4 0 200 100 T180
TEXT270 4 0 200 100 T270
FORM
PRINT
```

Output:



4. Bar Code Commands

BARCODE Command

The BARCODE command prints bar codes in both vertical and horizontal orientations at specified widths and heights.

Standard Bar Codes

Format:

{command} {type} {width} {ratio} {height} {x} {y} {data}

where:

{command}: Choose from the following:

BARCODE(or B): Prints bar code horizontally.

VBARCODE (or VB) Prints bar code vertically.

{type}: Choose from the following table:

Symbology:	**Use:**
UPC-A	UPCA
UPC-E	UPCE
EAN/JAN-13	EAN13
EAN/JAN-8	EAN8
Code 39	39
Code 93/Ext. 93	93
Code 128 (Auto)	128
Codabar	CODABAR

Note: Barcode data must be supplied in the {data} section and before the new line character sequence. Otherwise, printer may consume the next command as barcode data therby producing erroneous barcode and not properly executing the next command.

{width}: Unit-width of the narrow bar.

{ratio}: Ratio of the wide bar to the narrow bar. Refer to the table in Appendix “E” for appropriate settings.

0 = 1.5 : 1	20 = 2.0:1	26 = 2.6:1
1 = 2.0 : 1	21 = 2.1:1	27 = 2.7:1
2 = 2.5 : 1	22 = 2.2:1	28 = 2.8:1
3 = 3.0 : 1	23 = 2.3:1	29 = 2.9:1
4 = 3.5 : 1	24 = 2.4:1	30 = 3.0:1
	25 = 2.5:1	

Note: The ratios in the Appendix are suggested for best results; however, any ratio can be assigned.

{height}: Unit-height of the bar code.

{x}: Horizontal starting position.

{y}: Vertical starting position.

{data}: Bar code data.

BAR CODE Example

Input:

```
! 0 200 200 210 1
BARCODE 128 1 1 50 150 10 HORIZ.
```

```
TEXT 7 0 210 60 HORIZ.
VBARCODE 128 1 1 50 10 200 VERT.
VTEXT 7 0 60 140 VERT.
FORM
PRINT
```

Output:

BARCODE-TEXT Command

The BARCODE-TEXT command is used to label bar codes with the same data used to create the bar code. The command eliminates the need to annotate the bar code using separate text commands. The text will be centered below the bar code.

Use BARCODE-TEXT OFF (or BT OFF) to disable.

Format:

{command} {font number} {font size} {offset}

where:

{command}: BARCODE-TEXT (or BT)

{font number}: The font number to use when annotating the bar code.

{font size}: The font size to use when annotating the bar code.

{offset}: Unit distance to offset text away from the bar code.

BARCODE-TEXT Example

Input:

```
! 0 200 200 400 1
JOURNAL
CENTER
; Annotate bar codes using font 7 size 0
; and offset 5 dots from the bar code.
BARCODE-TEXT 7 0 5
BARCODE 128 1 1 50 0 20 123456789
VBARCODE 128 1 1 50 40 400 112233445
BARCODE-TEXT OFF
FORM
PRINT
```

Output:

123456789
112233445

PDF417 (PORTABLE DATA FILE)

Format:

```
{command} {type} {x} {y} [XD n] [YD n] [C n] [S n]
{data}
<ENDPDF>
```

where:

{command}: Choose from the following:

BARCODE (or B): Prints bar code horizontally.

VBARCODE (or VB): Prints bar code vertically.

{type}: PDF-417

{x}: Horizontal starting position.

{y}: Vertical starting position.

[XD n]: Unit-width of the narrowest element. Range is 1 to 32, default is 2.

[YD n]: Unit-height of the narrowest element. Range is 1 to 32, default is 6.

[C n]: Number of columns to use. Data columns do not include start/stop characters and left/right indicators. Range is 1 to 30; default is 3.

[S n]: Security level indicates maximum amount of errors to be detected and/or corrected. Range is 0
to 8; default is 1.

{data} Bar code data.

<ENDPDF>: Terminates PDF-417.

Note: The BARCODE-TEXT command does not work with the PDF-417 bar code type. Any desired human readable text must be entered seperately with the TEXT command as in the following example.

PDF417 Example

Input:

```
! 0 200 200 210 1
B PDF-417 10 20 XD 3 YD 12 C 3 S 2
PDF Data
ABCDE12345
ENDPDF
T 4 0 10 120 PDF Data
T 4 0 10 170 ABCDE12345
FORM
PRINT
```

Output:

PDF Data
ABCDE12345

QR Code

Format:
{command} {type} {x} {y} [M n] [U n]
{data}
<ENDQR>
where:
{command}: Choose from the following:
BARCODE (or B): Prints bar code horizontally.
VBARCODE (or VB): Prints bar code vertically.
{type}:QR
{x}: Horizontal starting position.
{y}: Vertical starting position.
[M n]: QR code model number. Range is 1 or 2. QR Code Model 1 is the original specification, while QR Code Model 2 is an enhanced form of the symbology. Model 2 provides additional features and can be automatically differentiated from Model 1. Model 2 is the recommended model and is the default value.
[U n]: Unit-width/Unit-height of the module.
Range is 1 to 32. Default is 6.
{data}: Describes information required for generating a QR code. See the following examples.
{data} includes some mode selection symbols in addition to actual input data character string. The type of the input data could be recognized automatically by printer software or set "manually". There is a separator (comma) between mode selection symbols and the actual data.
Data field format for Automatic data type selection:
<Error Correction Level><Mask No.><Data Input Mode (should be "A")>,<Data Character String>
Error Correction Level should be one of the following symbols:
H - Ultra high reliability level (Level H);
Q - High reliability level (Level Q);
M - Standard level (Level M);
L - High density level (Level L).
Mask Number may be omitted or have a value from 0 to 8:
None - Automatic selection of the mask by software;
From 0 to 7 – use mask with corresponding number (0 to 7);
8 - No mask.
Data field format for manual data type selection includes additional character mode symbols and has the following format:
<Error Correction Level><Mask No.><Data Input Mode (should be "M")>,
<Character Mode 1><Data Character String 1>, <Character Mode 2><Data Character String 2>,< : >< : >,<Character Mode n><Data Character String n>
Character mode symbols:
N – Numeric;
A - Alphanumeric;
Bxxxx – Binary Binary mode includes number of data characters (xxxx) represented by 2 bytes of BCD code.
K – Kanji
Different data fields (with their corresponding character mode symbols) are separated by commas.

When the input mode is set to Automatic the binary codes of 0x80 to 0x9F and 0xe0 to 0xFF cannot be set.

<ENDQR>: Terminates QR code.

Data Field Formatting Examples

Example 1

Error Correction Level: Standard level <M>

Mask No.: <None>

Input mode: Automatic setting <A>

Data: QR Code

The {data} field presentation for generating a QR code under the conditions above:

MA,QR Code

Example 2

Error Correction Level: Ultra high reliability level <H>

Mask No.: <0>

Input mode: Manual setting <M>

Character Mode: Numeric mode <N>

Data: 0123456789012345

The {data} field presentation:

H0M,N0123456789012345

Example 3

Error Correction Level: Standard level <M>

Mask: <None> (Automatic selection)

Input mode: Manual setting <M>

Character Mode: Alphanumeric mode <A>

Data: AC-42

The {data} field presentation:

MM,AAC-42

Example 4

Error Correction Level: High density level <L>

Mask No.: Automatic setting <None>

Input mode: Manual setting <M>

Character Mode: Alphanumeric <A>

Data: QR code

Character Mode: Numeric <N>

Data: 0123456789012345

Character Mode: Alphanumeric <A>

Data: QRCODE

Character Mode: Binary <B>

Data: qrcode

The {data} field presentation:

LM,AQRcode,N0123456789012345,AQRCODE,B0006qrcode

Note: The BARCODE-TEXT command does not work with QR code. Any desired human readable text must be entered separately with the TEXT command as shown in the following example.

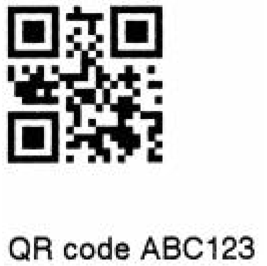
QR Code Example

Input:

```
! 0 200 200 500 1
B QR 10 100 M 2 U 10
MA,QR code ABC123
```

```
ENDQR
T 4 0 10 400 QR code ABC123
FORM
PRINT
```

Output:



QR code ABC123

NOTE: Human readable text is not part of the QR code output.

5. GRAPHICS

BOX Command

The BOX command provides the user with the ability to produce rectangular shapes of specified line thickness.

Format:

{command} {x0} {y0} {x1} {y1} {width}

where:

{command}: BOX

{x0}: X-coordinate of the top left corner.

{y0}: Y-coordinate of the top left corner.

{x1}: X-coordinate of the bottom right corner.

{y1}: Y-coordinate of the bottom right corner.

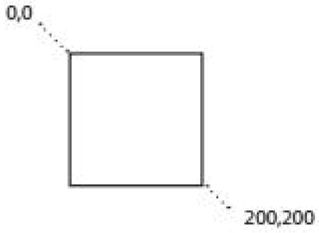
{width}: Unit-width (or thickness) of the lines forming the box.

BOX Command example

Input:

```
! 0 200 200 210 1
BOX 0 0 200 200 1
FORM
PRINT
```

Output:



Note: Text coordinates (in output) are shown for illustration purposes only.

LINE Commands

Lines of any length, thickness, and angular orientation can be drawn using the LINE command.

Format:

{command} {x0} {y0} {x1} {y1} {width}

where:

{command}: Choose from the following:

LINE (or L): Prints a line.

{x

0}: X-coordinate of the top-left corner.

{y0} Y-coordinate of the top-left corner.

{x

1}: X-coordinate of:

- top right corner for horizontal.
- bottom left corner for vertical.

{y1}: Y-coordinate of:

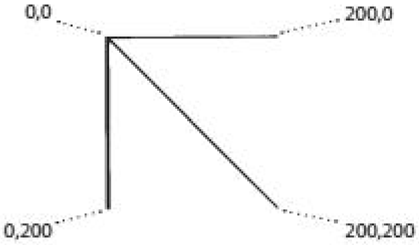
- top right corner for horizontal.
- bottom left corner for vertical.

{width}: Unit-width (or thickness) of the line

Input:

```
! 0 200 200 210 1
LINE 0 0 200 0 1
LINE 0 0 200 200 2
LINE 0 0 0 200 3
FORM
PRINT
```

Output:



Note: Text coordinates (in output) are shown for illustration purposes only.

INVERSE-LINE Commands

The INVERSE-LINE command has the same syntax as the LINE command. Previously created objects that lie within the area defined by the INVERSE-LINE command will have their black areas re-drawn white, and white areas re-drawn black. These objects can include text, bar codes and/or graphics, including downloaded .pcx files. INVERSE-LINE has no effect on objects created after its location, even if they fall within its covered area. In example INVERSE2.LBL, portions of the text field created after the INVERSELINE command remain black, hence invisible, even though placed within the INVERSE-LINE area.

Format:

{command} {x0} {y0} {x1} {y1} {width}

where:

{command}: Choose from the following:

INVERSE-LINE (or IL): Prints a line over an existing field to invert the image.

{x0}: X-coordinate of the top-left corner.

{y0}: Y-coordinate of the top-left corner.

{x1}: X-coordinate of:

- top right corner for horizontal.
- bottom left corner for vertical.

{y1}: Y-coordinate of:

- top right corner for horizontal.
- bottom left corner for vertical.

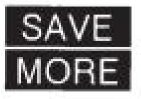
{width}: Unit-width (or thickness) of the inverse-line.

Inverse Line command examples

Input 1:

```
! 0 200 200 210 1
CENTER
TEXT 4 0 0 45 SAVE
TEXT 4 0 0 95 MORE
INVERSE-LINE 0 45 145 45 45
INVERSE-LINE 0 95 145 95 45
FORM
PRINT
```

Output 1:

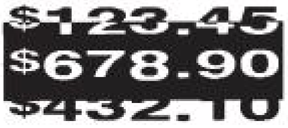


SAVE
MORE

Input 2:

```
! 0 200 200 210 1
T 4 2 30 20 $123.45
T 4 2 30 70 $678.90
IL 25 40 350 40 90
T 4 2 30 120 $432.10
FORM
PRINT
```

Output 2:



$123.45
$678.90
$432.10

GRAPHICS Commands

Bit-mapped graphics can be printed by using graphics commands. ASCII hex (hexadecimal) is used for expanded graphics data (see example). Data size can be reduced to one-half by utilizing the COMPRESSED-GRAPHICS commands with the equivalent binary character(s) of the hex data. When using CG, a single 8 bit character is sent for every 8 bits of graphics data. When using EG two characters (16 bits) are used to transmit 8 bits of graphics data, making EG only half as efficient. Since this data is character data, however, it can be easier to handle and transmit than binary data.

Format:

{command} {width} {height} {x} {y} {data}

where:

{command}: Choose from the following:

EXPANDED-GRAPHICS (or EG): Prints expanded graphics horizontally.

VEXPANDED-GRAPHICS (or VEG): Prints expanded graphics vertically.

COMPRESSED-GRAPHICS (or CG): Prints compressed graphics horizontally.

VCOMPRESSED-GRAPHICS (or VCG): Prints compressed graphics vertically.

{width}: Byte-width of image.

{height} Dot-height of image.

{x}: Horizontal starting position.

{y}: Vertical starting position.

{data}: Graphics data.

Graphics command example

Input:

```
! 0 200 200 210 1
EG 2 16 90 45 F0F0F0F0F0F0F0F00F0F0F0F0F0F0F0F
F0F0F0F0F0F0F0F00F0F0F0F0F0F0F0F
FORM
PRINT
```

Output:

Note: Graphic output has been magnifed. Actual size is 1/4 of output shown.

PCX Commands (For HM-T3 only)

The PCX command gives a user the ability to send ".PCX" graphics formatted images to the printer. The
.PCX image MUST be encoded as a black and white image.

Format:

```
{command} {x} {y}
{data}
```

where:

{command}: PCX
{x}: X-coordinate of the top-left corner.
{y}: Y-coordinate of the top-left corner.
{data}: PCX image data.

PCX Command example1

Input1:

In the example below, the image is sent in three steps. First, the printer is sent commands to expect a
.PCX formatted file. The second input to the printer is the .PCX image. This image must be a 2 color (black
and white) image. The last step is to tell the printer to print the label.

```
! 0 200 200 500 1
PCX 0 30
Input 2 (IMAGE.PCX)
Input 3 (ENDPCX.LBL)
FORM
PRINT
```

Output1:

PCX Command Example 2

In this example , the PCX image has been loaded into the printer's flash file system and given the name
"IMAGE.PCX". The "!<" operator can now be used to instruct the printer to get the data stored in the file
"Image.PCX" and use it for building the image.

Input 2:

```
! 0 200 200 500 1
PCX 0 30 !<IMAGE.PCX
FORM
PRINT
```

Output2:

6. ADVANCED COMMANDS

JUSTIFICATION Commands

Alignment of fields can be controlled by using the justification commands. By default, the printer will left justify all fields. A justification command remains in effect for all subsequent fields until another justification command is specified.

Format:

```
{command} [end]
```

where:

{command}: Choose from the following:

CENTER: Center justifies all subsequent fields.

LEFT: Left justifies all subsequent fields.

RIGHT: Right justifies all subsequent fields.

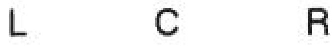
[end]: End point of justification. If no parameter is entered, justification commands use the printhead's width for horizontal printing or zero (top of form) for vertical printing.

JUSTIFICATION Example

Input:

```
! 0 200 200 210 1
CENTER 383
TEXT 4 0 0 75 C
LEFT
TEXT 4 0 0 75 L
RIGHT 383
TEXT 4 0 0 75 R
FORM
PRINT
```

Output:



```
L          C          R
```

PAGE-WIDTH Command

The printer assumes that the page width is the full width of the printer. The maximum height of a print session is determined by the page width and the available print memory. If the page width is less than the full width of the printer, the user can increase the maximum page height by specifying the page width.

Note: This command should be issued at the beginning of a print session.

Format:

```
{command} {width}
```

where:

{command}:N Choose from the following:

PAGE-WIDTH (or PW): Specifes page width.

{width}: Unit-width of the page.

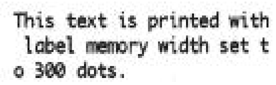
PAGE-WIDTH Examples

Input 1:

```
! UTILITIES
SETLP 7 0 15
PW 300
PRINT
```

This text is printed with label memory width set to 300 dots.

Output 1:



```
This text is printed with
 label memory width set t
o 300 dots.
```

PREFEED Command (For HM-T3 only)

The PREFEED command instructs the printer to advance the media a specified amount prior to printing.

NOTE: The PREFEED command instructs the printer to advance the media a specified amount prior to printing.

Format:

{command} {length}

where:

{command}: PREFEED

{length}: Unit length the printer advances media prior to printing.

PREFEED Command Example

The following example sets up the printer for pre-feeding 40 dot-lines prior to printing.

Input:

```
! 0 200 200 210 1
PREFEED 40
TEXT 7 0 0 20 PREFEED EXAMPLE
FORM
PRINT
```

PACE Command

This command can be used with batch printing. When PACE is activated, the user must depress the printer's "FEED" key to print additional labels until the batch quantity is exhausted. By default, pacing is disabled on power up.

Format:

{command}

where:

{command}: PACE

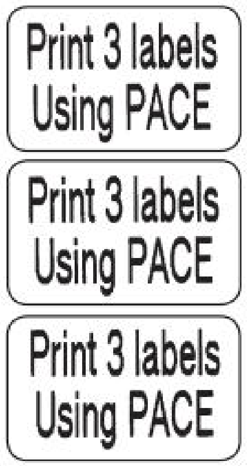
PACE Command Example

In the following example, the command fle shown was sent to the printer once. The two additional printouts were produced by pressing the 'FEED' key once for each additional printout.

Input:

```
! 0 200 200 210 3
; Tell printer to print a label
; after each 'FEED' key press
; until all 3 labels are printed
PACE
; Printer holds journal stock
JOURNAL
; Center the text
CENTER
TEXT 4 1 0 10 Print 3 labels
TEXT 4 1 0 90 Using PACE
PRINT
```

Output:



Print 3 labels Using PACE

Print 3 labels Using PACE

Print 3 labels Using PACE

WAIT Command

This command is used to introduce a delay after a label is printed.

Format:

{command} {delay-time}

where:

{command}: WAIT

{delay-time}: Delay time in 1/8 seconds.

WAIT Command Example

In the example below, the printer will pause 10 seconds (10 * 8 = 80) after printing each label.

Input:

```
! 0 200 200 150 5
WAIT 80
TEXT 5 0 0 20 DELAY 10 SECONDS
FORM
PRINT
```

TENSION Commands

The tension commands are used to adjust the liner tension before and/or after printing a label by running the rewind motor for a pre-specified length. This adjustment improves peeler performance of printers equipped with a motorized rewind mechanism. The TENSION commands are ignored by printers not equipped with a motorized rewind.

Format:

{command} {length}

where:

{command}: Choose one of the following:

PRE-TENSION: Perform tension adjustment prior to printing the label.

POST-TENSION: Perform tension adjustment after printing the label.

{length}: The unit length the rewind motor should to tighten the liner tension. The re-wind motor will slip once tension is adjusted (it will not pull the stock out of adjustment for the next print cycle).

TENSION Command Example

In this example, the printer is instructed to run the rewind motor for 30 dot-lines, removing any slack in the liner to adjust the liner tension prior to printing the label.

Input:

```
! 0 200 200 150 1
PRE-TENSION 30
TEXT 5 0 0 20 ADJUSTS TENSION
```

ON-OUT-OF-PAPER Command (For HM-T3 only)

This command can be issued to instruct the printer as to the course of action to take when it encounters

an error while printing the label (such as running out of stock)

Format:

{command} {action} {number of retries}

where:{ command}: ON-OUT-OF-PAPER

{action}: Choose one of the following:

PURGE: Discard the label if print error is encountered after the specified number of attempts.

WAIT: Do not discard the label if print error is encountered. In this mode the printer will wait for the error to be corrected before making the next print attempt.

{number of retries}: n: Specify how many times the printer should attempt to print the label.

The default printer configuration is: ON-OUT-OF-PAPER PURGE 2

ON-OUT-OF-PAPER Command Example

This example instructs the printer to attempt to print the label twice.

Input:

```
! 0 200 200 150 1
ON-OUT-OF-PAPER WAIT 2
TEXT 5 0 0 20 MAKE TWO ATTEMPT
FORM
PRINT
```

SPEED Command

This command is used to set the highest motor speed level. Each printer model is programmed with a minimum and maximum attainable speed. The SPEED command selects a speed level within a range of 0 to 5, with 0 the slowest speed. The maximum speed programmed into each printer model is attainable only under ideal conditions. The battery or power-supply voltage, stock thickness, print darkness, applicator usage, peeler usage, and label length are among the factors that could limit the maximum attainable print speed.

WARNING: By exercising this command the user overrides the factory programmed speed for the label being printed, which may adversely affect print quality. If print quality suffers using the current SPEED setting, the printer speed should be reduced.

Format:

{command} {speed level}

where:

{command}: SPEED

{speed level}: A number between 0 and 5, 0 being the slowest speed.

SPEED Command Example

Input:

```
! 0 200 200 150 1
SPEED 4
TEXT 5 0 0 20 PRINTS AT SPEED 4
FORM
PRINT
```

SETSP Command

The SETSP command is used to change spacing between text characters.

Format:

{command} {spacing}

where:

{command}: SETSP

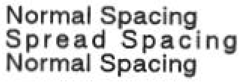
{spacing}: Unit measurement between characters. The default for spacing is zero. Note that this command is affected by the UNITS command setting.

SETSP CommandExample

Input:

```
! 0 200 200 210 1
T 4 0 0 10 Normal Spacing
SETSP 5
T 4 0 0 50 Spread Spacing
SETSP 0
T 4 0 0 90 Normal Spacing
FORM
```

Output:



```
Normal Spacing
S p r e a d  S p a c i n g
Normal Spacing
```

COUNTRY / CODE PAGE Command

The COUNTRY control command substitutes the appropriate character set for the specified country.

Format:

{command} {name}

where:

{command}: COUNTRY

{name}: Choose from the following table:

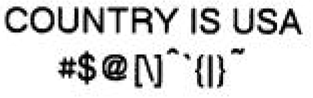
USA	GERMANY	FRANCE
SWEDEN	SPAIN	NORWAY
ITALY	CP850	UK
LATIN9	CP874 (Thai)	CHINA (Simplified Chinese, Double Byte Character Set- see Asian Fonts topic on Pg. 8-23)
KOREA (Korean, Double Byte Character Set- see Asian Fonts topic on Pg. 8-23)	BIG5 (Traditional Chinese, Double Byte Character Set- see Asian Fonts topic on Pg. 8-23)	JAPAN-S (S-JIS, Double Byte Character Set- see Asian Fonts topic on Pg. 8-23)

COUNTRY Command Example

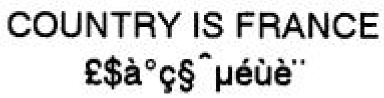
Input:

```
! 0 200 200 80 1
IN-MILLIMETERS
JOURNAL
CENTER
; Set the country as USA
COUNTRY USA
; Now Print Text From ISO substitution Table
TEXT 4 0 0 8 COUNTRY IS USA
TEXT 4 0 0 15 #$@[\]^‘{|}~
; Set country for France and print the same text
COUNTRY FRANCE
TEXT 4 0 0 28 COUNTRY IS FRANCE
TEXT 4 0 0 35 #$@[\]^‘{|}~
PRINT
```

Output:



```
COUNTRY IS USA
#$@[\]^`{|}~
```



```
COUNTRY IS FRANCE
£$à°ç§^µéùè¨
```

DEFINE FORMAT

Defining a label format file is accomplished using the DEFINE-FORMAT (or DF) command to mark the beginning of the format, and PRINT to mark the end. A '\\' (double-backslash) acts as a place holder for data.

DEFINE FORMAT Command Example

Input:

```
! DF SHELF.FMT
! 0 200 200 210 1
CENTER
TEXT 4 3 0 15 \\
TEXT 4 0 0 95 \\
BARCODE UPCA 1 1 40 0 145 \\
TEXT 7 0 0 185 \\
FORM
PRINT
```

USE-FORMAT

The USE-FORMAT (or UF) command instructs the printer to use a specified format file. The label will be created using that format file with data supplied following the USE-FORMAT command. After accessing the specified format file, the printer substitutes the '\\' delimiters with the data supplied, producing the desired label.

USE FORMAT Command Example

Input:

```
! UF SHELF.FMT
$22.99
SWEATSHIRT
40123456784
40123456784
```

As with all print commands, each line in a format file and its accompanying variables must be terminated with the carriage-return and line-feed character sequence.

Once defined, a format will remain in the printer's nonvolatile memory for future reference. An existing format can be changed by over writing the file. By using the DEL command, the format file can be deleted.

Format file names can consist of no more than 8 letters or digits, and format file extensions can be no more than 3 letters or digits. Any lowercase letter in the format file name or extension will be converted to upper case.

Note: Every time a fle is created on the printer using, for example, the "! DEFINE-FORMAT...", "! DF..." or the Label Vista application the fle information is written to flash memory. Unlike RAM, flash memory does not require battery for retaining data, and is immune to data corruption due to static discharge. Although flash memory is superior to RAM for safe-guarding fle contents, it is limited to an average of 10,000 write cycles (i.e. fle creations). For this reason, the user should exercise the fle creation commands such that the stated limit is not exceeded.

BEEP Command

This command instructs the printer to sound the beeper for a given time length. Printers not equipped with a beeper will ignore this command.

Format:

{command} {beep_length}

Where:

{command}: BEEP

{beep_length}: Duration of beep, specified in (1/8th) second increments.

BEEP Command Example

This example instructs the printer to beep for two seconds (16 x .125 seconds = 2 seconds)

Input:

```
! 0 200 200 210 1
CENTER
TEXT 5 0 0 10 beeps for two seconds
BEEP 16
FORM
PRINT
```

7. LINE PRINT MODE

SETLP Command

Selecting the line printer font (the SETLP command) will change the font the printer uses for line print mode. It also chooses the amount of space the printer will move down when the printer recieves a carriage return (hex value 0x0d).

! U1 SETLP {font name or number} {size} {unit height}

The {unit height} value should be set to the actual height of the font being used. Refer to Appendix C of this manual for actual resident font height values.

SETLP allows you to use either the resident fonts or pre-scaled fonts downloaded to the flash memory.

The Label Vista design software can create and upload a font for the printer from any available TrueType1 font. Appendix D contains a table of all resident font heights and their proper unit height.

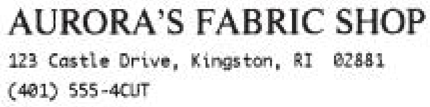
You can set the printer font multiple times when using the line printer to make a receipt. For example, to put the company name in a larger font at the top of a label, change to font 5 size 2 and then to font 7 size 0.

SET LP Command Example

Input:

```
! U1 SETLP 5 2 46
AURORA’S FABRIC SHOP
! U1 SETLP 7 0 24
123 Castle Drive, Kingston, RI 02881
(401) 555-4CUT
```

Output:



```
AURORA’S FABRIC SHOP
123 Castle Drive, Kingston, RI  02881
(401) 555-4CUT
```

SETLF Command

Use the SETLF command to change the height of each line without changing the font.

Format:

```
! U1 SETLF {unit height}
```

The command “! U1 SETLF 40” will advance the paper 40 dots for every LF (line feed, hex value 0x0a)

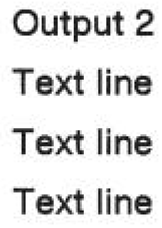
character it receives.

SETLF Command Example

Input:

```
! U SETLP 4 0 40
SETLF 40
PRINT
Output 2
Text line
Text line
Text line
```

Output :



```
Output 2
Text line
Text line
Text line
```

Moving With X and Y Coordinates

Even though the printer is in a line print mode, it can still move down and across the paper using X and Y values.

Format:

```
! U1 X {unit value}
! U1 Y {unit value}
! U1 XY {x unit value} {y unit value}
! U1 RX {unit x value to move relative to present position}
! U1 RY {unit y value to move relative to present position}
! U1 RXY{unit x value to move relative to present position} {unit y value to move relative to present position}
```

This command is useful for moving across the paper without using extra spaces or moving down the paper without needing to set the SETLF command to a specific value.

Negative values cannot be used for “Y” coordinates.

LMARGIN Command

The LMARGIN command sets the left margin in line print mode. Instead of issuing several X commands or inserting spaces, the LMARGIN command moves everything over the number of dots you choose.

Format:

```
! U1 LMARGIN {dots to offset from left}
```

This function can be used with the PAGE-WIDTH command. LMARGIN will move the left margin over the set number of dots from the automatically calculated side of the paper.

SETBOLD Command

The SETBOLD command will make text bolder and slightly wider. The SETBOLD command takes one operand to set how black the text should be made.

Format:

```
! U1 SETBOLD {value}
```

where {value} is an offset number from 0 to 5.

Notes: {value} will be in the units set by the UNITS command.

The default UNITS setting is in dots. (203 dots= 1”)

If UNITS is in inches the offset value range is 0-.0246”.

If UNITS is in centimeters the offset value range is 0-.0625 cm.

If UNITS is in millimeters the offset value range is 0-.625 mm.

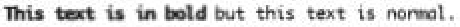
Be sure to issue a “! U1 SETBOLD 0” command to turn the bolding off when done.

SET BOLD Command Example

Input:

```
! U1 SETBOLD 2
This text is in bold ! U1 SETBOLD 0
but this text is normal.
```

Output:



```
This text is in bold but this text is normal.
```

Form Feed

ASCII Character (0x0c) will advance the paper to either the next index mark, or the length specified by the PAGE-HEIGHT, SETFF or SET-TOF commands. (The index mark is either a black line on the back of the stock, or the gap between labels. See GAP-SENSE or BAR-SENSE in Section 12 of this manual.)

Backspace

ASCII Character (0x08) acts as a non-destructive backspace. The character after the backspace character will appear on top of the previous character.

SETFF Command

The SETFF command is used to align top of media to printhead. Once this command is executed, the alignment will occur when :

- feed key is pressed.
- form-feed character (0x0c) is issued.
- FORM command is issued.

Format:

```
<!> <UTILITIES>
{command} {max-feed} {skip-length}
<PRINT>
```

where:

{command}: SETFF

{max-feed}: Maximum unit-length the printer advances searching for the next eye-sense mark to align top of form.

Valid values are 0-20,000.

{skip-length}: Unit-length printer advances past top of form. Valid values are 5-50.

SETFF Command Example

The following example programs the printer to advance the paper until the eye-sense mark is found, or the paper has been advanced a maximum of 25 millimeters. If an eye-sense mark is found, the paper will be advanced an additional 2.5 millimeters.

Input:

```
! UTILITIES
IN-MILLIMETERS
SETFF 25 2.5
PRINT
```

COUNTRY / CODE-PAGE Utility or CHAR-SET/CODE PAGE Utility (For HM-T3 only)

The COUNTRY or CHAR-SET utility command substitutes the appropriate character set for the specified
country. The two commands can be used interchangebly.

See Section 8, page 22 for more information on the COUNTRY or CHAR SET command.

Format:

```
<!> <UTILITIES>
{command} {name}
<PRINT>
```

where:

{command}: COUNTRY or CHAR-SET

{name}: Choose from the following:

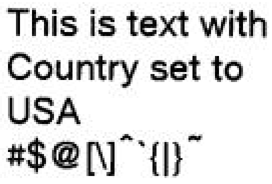
```
                    USA
                    GERMANY
                    FRANCE
                    SWEDEN
                    SPAIN
                    NORWAY
                    ITALY
                    CP850
                    UK
                    LATIN9
            CP874 (Thai)
            CHINA (Simplified Chinese, Double Byte Character Set)
            KOREA (Korean, Double Byte Character Set)
            BIG5 (Traditional Chinese, Double Byte Character Set)
            JAPAN-S (S-JIS, Double Byte Character Set)
```

Input:

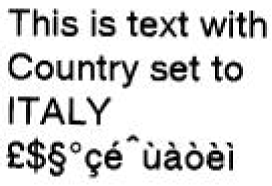
```
! UTILITIES
; Tell the printer to use font 4 size 0
; for line printer mode with 5 millimeters
; line spacing
IN-MILLIMETERS
SETLP 4 0 5
COUNTRY USA
; or CHAR-SET USA
PRINT
This is text with
Country set to
USA
#$@[\]^  ‘{|}~
! UTILITIES
COUNTRY ITALY
; or CHAR-SET ITALY
PRINT
```

This is text with
Country set to
ITALY
#$@[\]^ ‘{|}~

Output:



This is text with
Country set to
USA
#$@[\]ˆ`{|}˜



This is text with
Country set to
ITALY
£$§°çéˆùàòèì

Note: The printer must be configured with fonts that contain the extended character sets used in the selected country.

8. ADVANCED UTILITIES

VERSION Utility

This command reports the firmware version as a four character null-terminated ASCII string.

Format:

```
<!> <UTILITIES>
{command}
<PRINT>
```

where:

{command}: VERSION

VERSION Example

Input:

```
! UTILITIES
VERSION
PRINT
```

TIMEOUT Utility

The TIMEOUT command allows you to set the time the printer will remain on without receiving data. If no data is received after the specified timeout, the printer will turn itself off to save energy and preserve battery life. You can disable the timeout feature by setting the timeout value to 0.

Format:

```
<!> <UTILITIES>
{command} {time}
<PRINT>
```

where:

{command}: TIMEOUT

{time}: Time in 1/8 seconds of inactivity before printer will turn itself off.

TIMEOUT Utility Example

Input :

```
! UTILITIES
TIMEOUT 960
PRINT
```

This example sets the printer to turn off after 2 minutes of inactivity (120 seconds X 8 = 960).